

Universal Serial Bus PC Legacy Compatibility Specification

**0.9 Draft Revision
May 30, 1996**

Scope of this Revision

The 0.9 release candidate of this definition is intended for general industry review.

Technical Editor

Tom Roden Phoenix Technologies
Tom_Roden@PTLTD.COM

Contributors

AMI Prashant Sethi

Award Maurice Bizzarri, Jim Busse, Jeff W. Chen

CMD Mike Chen

Compaq Dave Wooten

Digital Equipment Corp. Tom Schmidt

IBM Michael F. Bucalos

Intel Robert P. Hale, Brad Hosler

NEC Masami Katagiri

Phoenix Mike Goodman, Tom Roden, Curtis Stevens

SystemSoft James Kelsey, David Lawrence

Revision History

Rev	Date	Filename	Author	Comments
0.7	1/25/96	LEGR1B.DOC	Thomas A. Roden	Initial Release
0.7.2	1/25/96	LEGR1C.DOC	Thomas A. Roden	Incorporated comments
0.7d	3/26/96	LEGR1D.DOC	Thomas A. Roden	Incorporated comments from February 15 Face to Face
0.8	4/18/96	LEGR2.DOC	Thomas A. Roden	Incorporated comments from March 28 Face to Face
0.8a	4/30/96	LEGR2a.DOC	Thomas A. Roden	Incorporated comments from April 22 Face to Face
0.9	5/30/96	usb_le9	Thomas A. Roden	Updated version text for 0.9 general release

USB PC Legacy Compatibility Specification
Copyright © 1996, USB Implementers Forum
All rights reserved.

INTELLECTUAL PROPERTY DISCLAIMER

THIS SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

A LICENSE IS HEREBY GRANTED TO REPRODUCE AND DISTRIBUTE THIS SPECIFICATION FOR INTERNAL USE ONLY. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY OTHER INTELLECTUAL PROPERTY RIGHTS IS GRANTED OR INTENDED HEREBY.

AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF PROPRIETARY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. AUTHORS OF THIS SPECIFICATION ALSO DO NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATION(S) WILL NOT INFRINGE SUCH RIGHTS.

All other product names are trademarks, registered trademarks, or servicemarks of their respective owners.

Please send comments via electronic mail to usbdevice@fsp008.fm.intel.com

Table of Contents

1. Introduction.....	1
1.1 Scope	1
1.2 Purpose	1
1.3 Conventions.....	1
1.3.1 PnP Dependencies	1
1.4 Related Documents	1
1.5 Terms and Abbreviations	2
2. Management Overview.....	3
3. Reference Architectures	4
3.1 Legacy With No USB Support.....	4
3.2 USB Legacy Mode Support With Keyboard Controller.....	4
3.3 USB Legacy Mode Support With Keyboard Controller Emulator	5
3.4 USB Extrinsic Support With Keyboard Controller	5
3.5 Issues Not Covered In Reference Architectures	6
4. Functional Characteristics	6
4.1 Operational Model	6
4.1.1 Legacy Mode Support.....	6
4.1.2 Extrinsic Support	6
4.1.3 Transitioning From Legacy Mode To Extrinsic Support.....	6
4.1.4 Transitioning From Extrinsic To Legacy Mode Support.....	7
4.2 Interfaces.....	7
4.2.1 Host Controller	7
4.2.2 Motherboard USB Legacy Mode Support Hardware.....	7
4.2.3 Motherboard Emulation Hardware -- Keyboard Controller	7
4.2.4 Motherboard Emulation Hardware -- Keyboard Controller Emulator	8
4.3 Functional Characteristics of Emulated Devices	8
4.3.1 Multiple Keyboard/Mouse Emulation.....	8
4.3.2 Hot Attach/Detach	8
5. System Requirements.....	9
5.1 Host Controller.....	9
5.2 Motherboard USB Legacy Mode Support Control.....	10
5.3 Motherboard Emulation Hardware -- Keyboard Controller	10
5.4 Motherboard Emulation Hardware -- Keyboard Controller Emulator	10
5.5 USB KBC Compatibility Test Coverage	10

5.6 KBC Input Buffer Full (IBF) bit clears within 100mS.....	11
5.7 KBC Output Buffer Full (OBF) bit clears after up to 100 reads spaced 1mS apart <how generated?>	11
5.8 KBC Gate A20 command causes 0x0/0x100000 address aliasing when set low, but not when set high (tested 100 times).....	11
5.9 KBC Processor Reset command causes transition to real mode and transfer of control to address stored at BIOS Data Area[067H].....	11
5.10 KBC OBF going true causes IRQ1 to occur (tested ??? times).....	11
5.11 KBC echoing “.” causes INT15H/4fH vector to produce “.” value (tested ??? times)	11
5.12 KBC echoing “.” causes INT16H to produce “.” value (tested ??? times)	11
5.13 Set keyboard LEDs and verify no errors reported by keyboard (tested ??? times)	11
6. USB Support Transition Sequences	12
6.1 USB Control Transition Algorithms.....	12
6.1.1 Transition To Extrinsic Support.....	12
6.1.2 Transition To Legacy Mode Support	12

List of Tables

Rev ii

Date..... ii

Filename ii

Author ii

Comments ii

1. Introduction

The Universal Serial Bus (USB) is a communications architecture that gives a PC the ability to interconnect a variety of devices via a simple four-wire cable. The USB is actually a two-wire serial communication link that runs at 12 megabits (Mbs) per second. USB protocols can configure devices at startup or when they are plugged in at run time.

USB allows for the replacement of peripherals that have previously been attached to special-purpose interfaces and accessed by direct-hardware methods. This raises software compatibility issues that must be addressed. The point in system boot or the types of operating environments requiring support determines when support must be installed and what interfaces it must provide.

Support for devices such as USB keyboard may be required before operating system (O/S) support can be loaded. In addition, legacy O/Ses may not readily support USB drivers. Compatibility with legacy devices offers significant commercial value.

1.1 Scope

The scope of this document is limited to keyboard and pointing device support for legacy compatibility. Briefly, this document describes:

- Reference architectures that provide for PC legacy compatibility - especially in the areas of keyboard and pointing device support
- System requirements that allow the transfer between legacy mode (i.e. BIOS) and extrinsic (i.e. O/S) USB device support

1.2 Purpose

This document assists manufacturers of PCs in making systems that have a well-defined minimum level of PC legacy compatibility. By providing reference architectures, complete solutions are presented in an easy-to-understand fashion. These solutions are intended to function in non-USB-aware operating environments, and testing criteria are provided to validate minimal functionality.

By providing the requirements for transferring between legacy mode and extrinsic USB device support, legacy mode support can be designed that does not interfere with the USB-aware operating environments.

1.3 Conventions

1.3.1 PnP Dependencies

In this document, PnP is described as the method for controlling motherboard device configuration because it is the currently accepted standard for this type of operation. In the future when more advanced methods become available, these will become the recommended methods with regards to USB operations as well.

1.4 Related Documents

The following specifications are referenced by this document:

- *USB Support for PC Legacy Software*, Revision <0.?.>. USB Implementers Forum.

- *Universal Serial Bus Specification*, 1.0 (also referred to as the USB Specification). In particular, see Chapter 9, “USB Device Framework.”
- *Human Input Devices (HID) Class Definition*, Revision <0.?.>. USB Implementers Forum.
- *Plug and Play BIOS Specification*, 1.0a. Compaq, Phoenix, Intel.
- *ACPI Specification* <name?>, <Rev?> <Author?>.
- *PS/2-Model 80 Technical Reference Manual*, First Edition, April 1987. IBM.
- *IBM Personal System/2 Mouse Technical Reference*, Second Edition, June 1989. IBM.
- *Intel 8242BB, Technical Reference Manual*, Version 1.0, August 15, 1994, Intel.

1.5 Terms and Abbreviations

Auxiliary

Port Secondary serial port on KBCs. Usually used for pointing devices (e.g. mouse).

Extrinsic Support Support not inherent to a system. In USB terms, device support not specific to any particular computer system. Extrinsic support is generally part of a USB-aware operating environment.
An example of extrinsic support is the keyboard and mouse services of a virtualizing operating system such as Windows 95.

KBC KeyBoard Controller. A device that manages the traffic to and from AT or PS/2 keyboards and mice.

Legacy Any system characteristic or behavior accepted as industry compliant prior to the introduction of USB. Especially direct hardware register accesses.

Legacy Mode

Support Support inherent to a system. In USB terms, USB device support built into, or specific to, a particular computer system. Such support is generally provided by emulating legacy hardware interfaces. Legacy mode support allows devices to function in an operating environment that is not USB aware.
An example of legacy mode support is support for a USB keyboard and a USB pointing device that resides in the system BIOS and presents the keyboard and mouse data via the port 60H/64H/IRQ1/IRQ12 interface.

Operating

Environment Software controlling system level hardware and services. Normally an operating system, but may be a complex program containing operating system functionality.

POST Power On Self Test. The initial program loaded by a PC from its firmware. POST normally consists of hardware validation, motherboard device configuration, support installation, SETUP (the user interface for motherboard device configuration and support installation), and Option ROM scan (for external device configuration, support installation, and any associated user interfaces). POST terminates upon a successful Initial Program Load from media.

2. Management Overview

USB devices, such as keyboards, will replace their PS/2-style counterparts over the next few years. There is a large installed base of software that relies on the legacy functionality of PS/2 style devices.

To ease the transition, the USB PC Legacy Working Group is authoring this specification and an accompanying whitepaper that present our proposed solutions and the decision making process that led to them. The whitepaper is written for a much broader audience than this specification and should be read before this document.

Briefly, the whitepaper states that hardware register-level emulation is required to provide a solution compatible with the current installed base of PC software. For keyboard, mouse, and KBC emulation, this consists of emulation of I/O ports 60H and 64H and IRQ1 and IRQ12.

The following are goals of the support outlined in this document:

- To support the majority of legacy software applications and operating systems.
- To support USB-KBCTEST, a diagnostic to be written by this working group.
- To provide a minimal functionality set. See section 5.5 for a definition of this set.

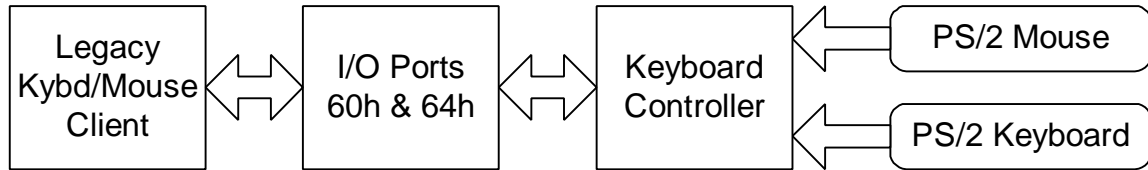
Despite the best efforts of the implementers, this group acknowledges that some compatibility may not be available with the solution described in this document. Timing and latency changes may disrupt the operation of some software in the installed base. Most notably, diagnostic software and certain TSR programs may be impacted. On the other hand, the vast majority of productivity and entertainment software is expected to work properly.

Note that this is a proposed solution, but is not intended to prevent OEMs from creating their own solutions with different feature sets and levels of support.

3. Reference Architectures

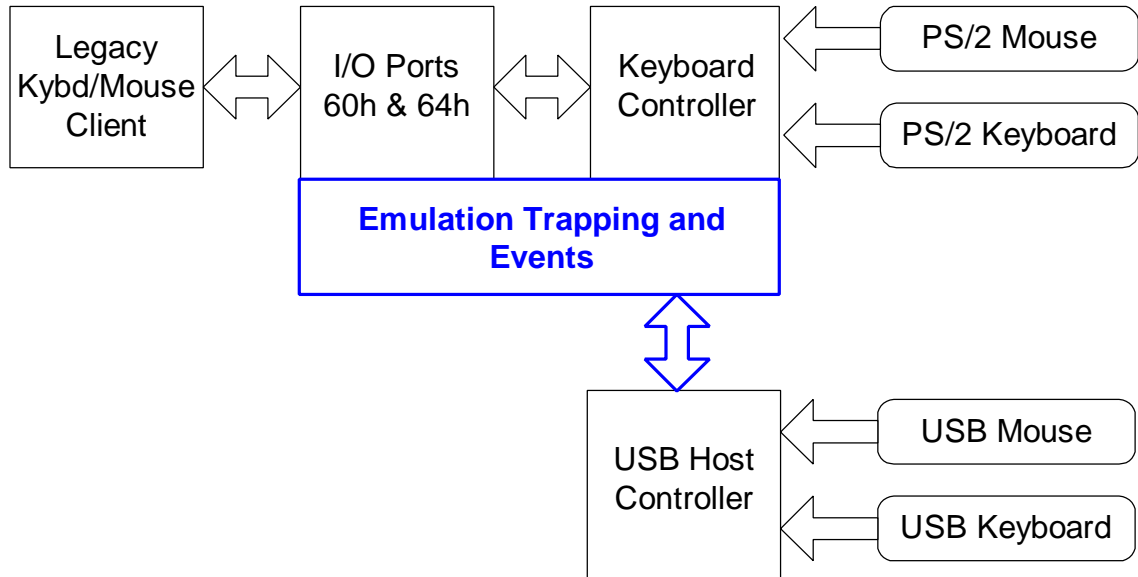
Note: The reference architectures in this section are provided for demonstration purposes only. They are not intended to specify a particular implementation or level of support required.

3.1 Legacy With No USB Support



The diagram above is a legacy system with no USB legacy compatibility.

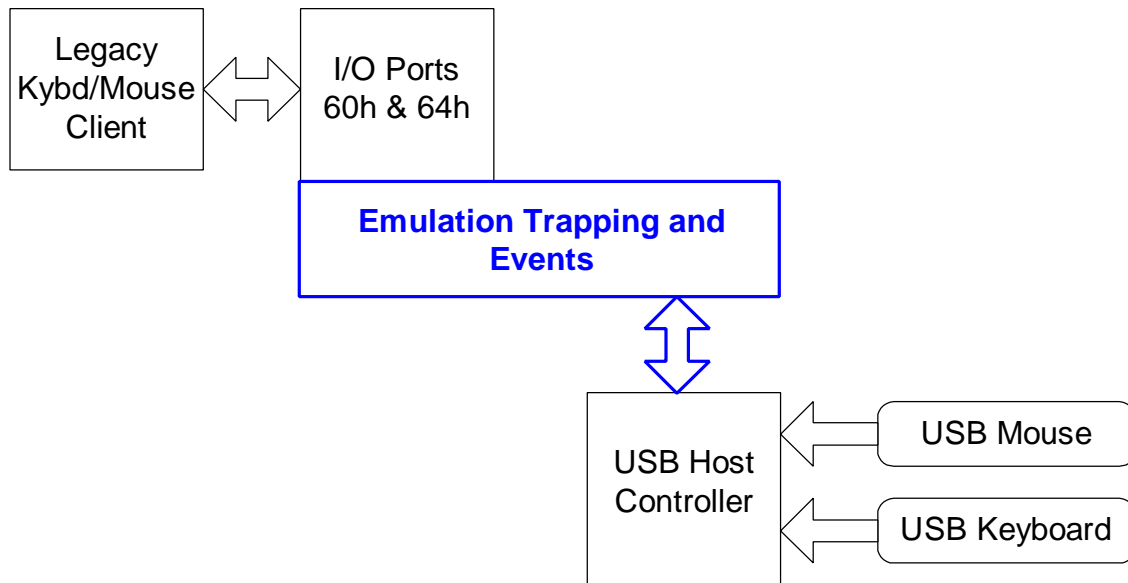
3.2 USB Legacy Mode Support With Keyboard Controller



The diagram above shows a USB system with a keyboard controller. Write accesses to the KBC cause emulation trapping. This trapping allows KBC commands to be propagated to USB peripherals when appropriate.

Emulation events are generated when the USB host controller has completed a transaction. Part of the service of emulation events is to insert data into the KBC. The KBC then generates the appropriate interrupt and makes the data available on the I/O bus.

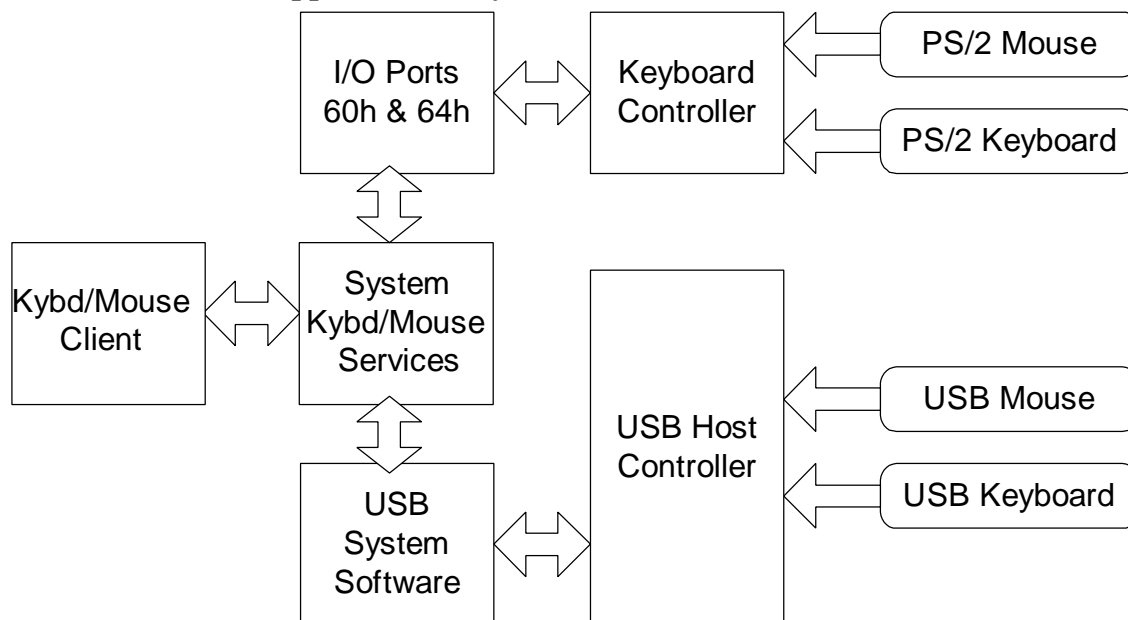
3.3 USB Legacy Mode Support With Keyboard Controller Emulator



The diagram above shows a system with a keyboard controller emulator. As in the first model, write accesses to the KBC emulator cause emulation trapping.

Emulation events function similarly, except that there are no PS/2 peripherals and no KBC. Emulation software and minimal emulation hardware are responsible for generating the appropriate IRQ and presenting the data to the CPU.

3.4 USB Extrinsic Support With Keyboard Controller



The diagram above shows a USB system with a keyboard controller in extrinsic mode. System software manages keyboard, mouse, and KBC traffic and directs it to the appropriate driver and/or hardware.

3.5 Issues Not Covered In Reference Architectures

It should be noted that keyboard controllers provide system controller functionality beyond keyboard and mouse support. Examples of this functionality are processor reset and Gate A20 operations. These are generally required for normal system operation. Defining this functionality is beyond the scope of this document, but the related documents do contain such specifications.

4. Functional Characteristics

This section describes the functional characteristics of a computer equipped with PC legacy software support, including:

- The operational model
- Interfaces
- Functional characteristics of emulated devices

4.1 Operational Model

4.1.1 Legacy Mode Support

Legacy mode support is normally loaded before any user input is accepted. Typically, legacy mode support resides within the BIOS and provides input device services for SETUP, Option ROM scan, and any other user POST control.

After POST terminates, legacy mode support provides device services for emulated devices. This typically means keyboard and optionally pointing device emulation via the hardware interface specified in *PS/2-Model 80 Technical Reference Manual*, *IBM Personal System/2 Mouse Technical Reference*, and *Intel 8242BB, Technical Reference Manual*. Additionally, KBC system control functionality must not be significantly impaired or must be emulated to the satisfaction of the client software. Legacy mode support remains in force until extrinsic support is installed, typically as part of a USB-aware operating environment.

4.1.2 Extrinsic Support

Once software has assumed control of the host controller and disabled any emulation hardware, it is free to support USB devices in any way. Details of extrinsic support are operating environment specific and are not part of this specification.

4.1.3 Transitioning From Legacy Mode To Extrinsic Support

To transition to extrinsic support, the O/S ceases accessing potentially emulated peripherals, locates a USB host controller, and performs the transition method specified by the host controller manufacturer. This method will not cause a deadlock because it is designed to stop asynchronous background events (i.e. HC service interrupts) before disabling the rest of the emulation hardware.

Once the transition method has been performed, the O/S installs its own device support.

4.1.4 Transitioning From Extrinsic To Legacy Mode Support

To transition to legacy mode support, the O/S removes device support, locates a USB host controller, and performs the transition method specified by the host controller manufacturer. This method will not cause a deadlock because it is designed to enable emulation hardware before restarting asynchronous background events.

Legacy mode support is not required to handle USB reconfiguration. This method must report whether it was successful in fully restoring legacy mode support. If not, the system must reboot to return to legacy mode support.

4.2 Interfaces

The rest of this section describes the interfaces recognized in connection with PC legacy software support.

4.2.1 Host Controller

The specification of the interfaces employed by the host controller is beyond the scope of this document. A section specifying the requirements of host controller interfaces appears later in this document.

4.2.2 Motherboard USB Legacy Mode Support Hardware

Any motherboard hardware, aside from the host controller, required to support legacy mode must be hidden from the system or marked unavailable. For example, schedule RAM must not be in the ranges normally scanned by existing operating systems, non-PCI I/O ranges must be contained in device nodes to prevent configurable devices from colliding with them.

The only information used for the transition is that about the host controller. Any required changes to support hardware will have to be made by emulation software as it detects a mode transition. These may include hiding device nodes and disabling the hardware they control. Host controller transition methods must take this into account by performing emulated operations as part of their published methods.

4.2.3 Motherboard Emulation Hardware -- Keyboard Controller

A computer system may implement legacy mode support using a keyboard controller to handle issues such as I/O buffering and IRQ generation. Keyboard controllers may allow a system designer to handle PS/2 keyboard and mouse hardware if present. A standard keyboard controller device node (e.g. PnP ID -- PNP0303) is sufficient to indicate the presence of the hardware. Similarly, if mouse IRQ generation is supported, a standard device node (e.g. PnP ID -- PNP0F13) will indicate its presence.

These device nodes may be disabled if they control no physical hardware. This capability is indicated via the existing PnP device node attribute bit field where bit 0 is set to 1 to indicate the device can not be disabled. If a device indicates it can be disabled, the existing PnP method for disabling a device node can be used. The extrinsic support software must determine if this hardware can be disabled (i.e. are actual KBC devices present) and disable it in the appropriate order (i.e. the mouse node should be disabled before the keyboard node).

There may also be the case in which the KBC may release its IRQ, but still requires its I/O space. This is indicated by the device node. It is not a problem since the typical I/O locations used (60H/64H) are not hotly contended resources.

4.2.4 Motherboard Emulation Hardware -- Keyboard Controller Emulator

To provide seamless emulation, the keyboard controller emulator will have to appear identical to standard device nodes (i.e. PNP0303 and PNP0F13). The only way to positively determine if an actual KBC is present is to enumerate the device nodes after the transition to extrinsic mode. If no KBC is present, the legacy device nodes may have been replaced by keyboard and mouse emulation device nodes with different device node numbers (PnP IDs -- TBD), or they may have simply been disabled or hidden. Alternately, an O/S could attempt to communicate with devices behind the KBC and these operations must time out. This does not determine KBC presence, but it does indicate whether PS/2 devices are present.

Since the keyboard controller emulator is not needed for extrinsic support, all emulation hardware device nodes should be disableable. It is recommended that extrinsic support software disable (or at least free the IRQs of) all such nodes to free system resources.

4.3 Functional Characteristics of Emulated Devices

The rest of this section describes functional characteristics of each emulated device type supported.

4.3.1 Multiple Keyboard/Mouse Emulation

With USB, there may be multiple keyboards and pointing devices attached to a system, especially if a legacy keyboard and/or auxiliary port connector is present. The following rules describe an acceptable legacy mode support behavior under these circumstances:

- If present, support for legacy keyboard and pointing device is required.
- Legacy mode support for USB keyboard is optional.
- Legacy mode support for USB pointing device is optional.

There are many consequences of the rules above. Most importantly, if a legacy keyboard connector is present, a system is guaranteed to support the legacy keyboard. This provides every user with a “safe mode” of operation that applies to all systems; attach a legacy keyboard and the system will work.

Since systems with no legacy mode support will be made, this support is specified as optional. In practice, however, any system with no PS/2 connectors would have to have legacy mode support for at least one USB keyboard or its usefulness would be limited.

4.3.2 Hot Attach/Detach

Legacy mode support is not required to handle USB reconfiguration. If devices are installed or removed during legacy mode support, the results are implementation dependent.

5. System Requirements

All systems with USB legacy mode support must have the following features:

- **Host Controller Data Routing** -- A mechanism must exist by which data from USB peripherals (the transfer of which is signaled by a host controller interrupt) can be routed to a legacy interface (i.e. a keyboard controller or keyboard controller emulator).
- **Legacy Mode Device Assignment** -- There is no defined mechanism for an operating system to determine which device or devices were involved in legacy mode support. The operating system is free to reassign devices for its own purposes.

There are further specific requirements for each element of the PC Legacy Compatibility System:

5.1 Host Controller

- **Positive Identification** -- All legacy mode support functions supported by the host controller must be implied by a unique identifier. For PCI host controllers, this would be the 32-bit PCI ID and/or the PCI class code. If PnP (i.e. device node based) host controllers are ever developed, their identifiers will be 32-bit PnP ID and/or device type/subtype/interface type. Once identified, the following methods are available:
 - **Definite Legacy Mode Detection Method** -- All uniquely identified implementations must be able to determine if legacy mode support is in force. If legacy mode support is unavailable, it is shown as never in force. The identification implies the mode detection method. Any host controller that is identified identically to another host controller must support the same mode detection method.
 - **Definite Legacy Mode To Extrinsic Transition Method** -- All uniquely identified implementations must be able to handle the transitions from legacy mode to extrinsic support and back in a manner understood by the extrinsic support software. Any host controller that is identified identically to another host controller must support the same mode transition method. Host controllers that report that they are in legacy mode must support the transition to extrinsic support. This method must return a report of which, if any, legacy resources (IRQ or I/O) were release by the transition to extrinsic mode. The format of that report is specified by the host controller vendor.
 - **Definite Extrinsic To Legacy Mode Transition Methods** -- All uniquely identified implementations must have a defined transition from extrinsic support to legacy mode in a manner understood by the extrinsic support software. Any host controller that is identified identically to another host controller must support the same mode transition method. If a host controller or its implementation in a system does not support this transition, it returns failure when this transition is attempted.
 - **Documented Identification and Mode Transition Methods** -- The host controller documentation must include the algorithms to positively identify the host controller and to perform the supported mode detection and transitions.
 - **Definitely Sequenced Mode Transition Methods** -- The mode transition methods must be sequenced to avoid deadlocking themselves. That is to say, if the software performing the transition has ceased accessing legacy mode supported devices and it executes the transition method, no background activity will occur that relies on hardware that has already been disabled.

- **Hardware-Based Methods** -- The mode detection and transition methods must be hardware based. This allows the operating system to execute the methods from its preferred processor mode.
- All other host controller issues are implementation specific. They may need to be known by the legacy mode support software, but they are beyond the scope of this document.
- Note that host controllers that don't implement legacy mode support are a degenerate case. Their transitions between legacy mode and extrinsic support states is a NULL operation.

5.2 Motherboard USB Legacy Mode Support Control

- Motherboard USB Legacy Mode Support Control is inferred from the host controller and is controlled by the methods that are bound to the host controller.

5.3 Motherboard Emulation Hardware -- Keyboard Controller

- Motherboards that provide legacy mode support via a keyboard controller must implement the interface specified in section 4.2.3.
- Any decode of KBC I/O locations for purposes other than KBC chip select (e.g. emulation trapping) is not controlled by KBC or mouse device nodes. This function would be part of the host controller transition methods.

5.4 Motherboard Emulation Hardware -- Keyboard Controller Emulator

- Motherboards that provide legacy mode support via a keyboard controller emulator must implement the interface specified in section 4.2.4.
- Any decode of KBC I/O locations for purposes other than KBC chip select (e.g. emulation trapping) is not controlled by KBC or mouse device nodes. This function would be part of the host controller transition methods.

5.5 USB KBC Compatibility Test Coverage

KBCTEST.EXE is a diagnostic utility to assure a basic level of functionality. Source and executable for this program will be available from USB developers forum. This utility performs the following tests:

5.6 KBC Input Buffer Full (IBF) bit clears within 100mS

5.7 KBC Output Buffer Full (OBF) bit clears after up to 100 reads spaced 1mS apart <how generated?>

5.8 KBC Gate A20 command causes 0x0/0x100000 address aliasing when set low, but not when set high (tested 100 times)

5.9 KBC Processor Reset command causes transition to real mode and transfer of control to address stored at BIOS Data Area[067H]

5.10 KBC OBF going true causes IRQ1 to occur (tested ??? times)

5.11 KBC echoing “.” causes INT15H/4fH vector to produce “.” value (tested ??? times)

5.12 KBC echoing “.” causes INT16H to produce “.” value (tested ??? times)

5.13 Set keyboard LEDs and verify no errors reported by keyboard (tested ??? times)

6. USB Support Transition Sequences

The following transition sequences are supplied to demonstrate the purpose of the system requirements and to ensure that there is agreement between the PC manufacturers and the O/S manufacturers about the specifics of the transitions to be supported.

6.1 USB Control Transition Algorithms

The transition to extrinsic support is performed by ceasing the accessing of potentially emulated peripherals, locating a USB host controller, and performing the transition method specified by the host controller manufacturer. This method will not cause a deadlock because it is designed to stop asynchronous background events (i.e. HC service interrupts) before disabling the rest of the emulation hardware.

USB Control Transitions must be performed when potentially emulated devices are not being polled. The following algorithms must be followed to assure successful control transitions on generic hardware platforms:

6.1.1 Transition To Extrinsic Support

- Cease accessing devices that may be emulated.
- For each host controller enumerated:
 - Perform any host controller specific operations required to detect state of legacy mode support
 - Perform any host controller specific operations required to disable legacy mode support.
- If resource reclamation is desired:
 - Re-enumerate PnP device nodes.
 - If a mouse device node (PnP ID -- PNP0F13) is present and it can be disabled, test for actual PS/2 mouse presence and disable device node if none found.
 - If a mouse emulation device node (PnP ID -- TBD2) is present and it can be disabled, disable device node.
 - If there is no enabled mouse or mouse emulation node present:
 - If the KBC device node is present and can be disabled (or its IRQ deallocated); test for actual keyboard presence and disable device node (or deallocate IRQ) if none found.
 - If a KBC emulation device node (PnP ID -- TBD1) is present, disable device node.

At this point, legacy mode support has been fully disabled and the operating environment is free to install extrinsic support. The specifics of this support is beyond the scope of this document.

6.1.2 Transition To Legacy Mode Support

There is no requirement for legacy mode support to re-enumerate the USB on transition to legacy mode support. This means that depending on attach/detach operations that occurred during extrinsic support, USB peripherals may not function after a return to legacy mode support. For this reason, environments may want to perform the transition to legacy mode support via a reboot and re-execution of POST. Additionally, host controller support for this transition is not guaranteed to be available.

For those environments that would like to attempt a transition to legacy mode support, follow these steps.

- Cease accessing devices that may be emulated.
- Restore the enable state of any KBC or mouse device nodes to that detected on entry.
- For each host controller enumerated:
 - Perform any host controller specific operations required to restore legacy mode support to that detected on entry.